

Moorhuhn

UI - Scribble Canvas - Winterlandschaft
Moorhuhn Spieldoberfläche

timer

1 Funktionale Analyse

Nutzererlebnis:

- spaß
- Herausforderung
- leichte Bedienung

Plattformwahl:

PC, da:
- größerer Bildschirm
- Shift Maus bedienbar
↳ höhere Treffsicherheit
(- Genauigkeit)

Spieldesign:

- Bei Betätigung der Leertaste wird Futter gestreut
- Mit Click der linken Maustaste wird ein Schneeball geworfen
- 20 Sek. Zeit, um so viele Vögel wie möglich abzuschüpfen
- Empfehlung: mit Maus spielen



Moorkuhlen - Funktionale Analyse I - First thoughts

1 Festlegung der Plattform: PC (Mobil) soll das eine Gegenüberstellung sein
 da - größeres Bildschirm
 - Weit Sicherheit größer - je nach Zeit mit Raum

Ausicht

• DB?

- * hier sollte es mit Maus spielen
- * bei click füllt Kuh auf Doden
- Reaktion auf Kuhstiere:

1200 x 550

16)



①/

Ende



▷ Start

dag



▷ Start

1 - - -
 2 - - -
 3. - - -
 4 - - -

noch Punktballen Sackeff

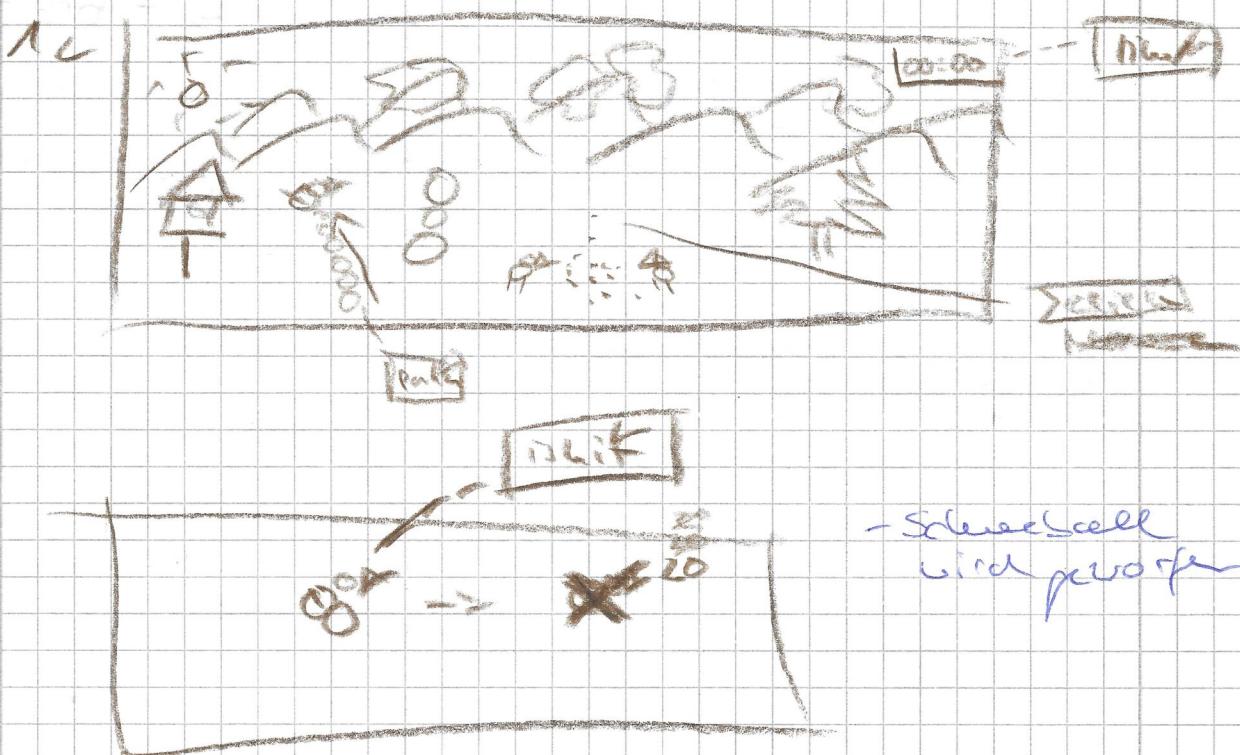
für rechts in
Säge 20/
in Säge 15?

Name: _____

Erreichte Punktzahl: _____



Formular
(MPJ + G)



- Schneeflocke
wird verbraucht

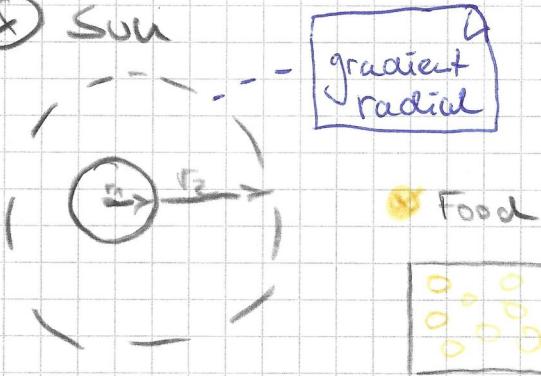
Info, die an Nutzer gegeben:

- Abstand: = Maerchen X - Dose Baby Powder
- Lebenszeit
- Tiere
- Punktzaehler ist an Ende gezeigt

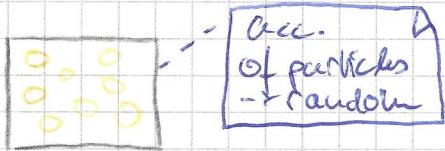
Interaktionen zwischen voneinander:

- Vögel \rightarrow Röhrchen wenn Futter hineingeworfen
Reaktion: ~~Vogel~~ \rightarrow 1. Vogel über dem Vorspielplatz \rightarrow ein Setzen
2. Vogel fügt Vorspiel
- Vögel \leftrightarrow Schneeflocke Vom Jenseite verschwinden
 \hookrightarrow Punktzaehler steht hier den

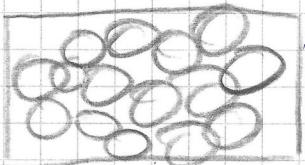
① Sun



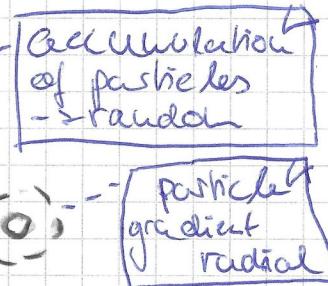
Food



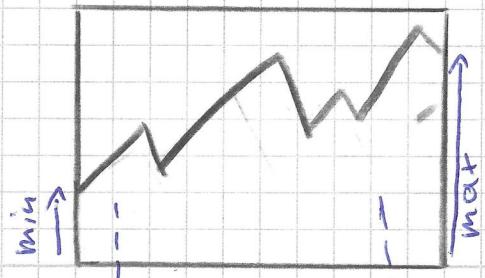
② Clouds



③



④ Mountains



random height between min and max

gradient linear

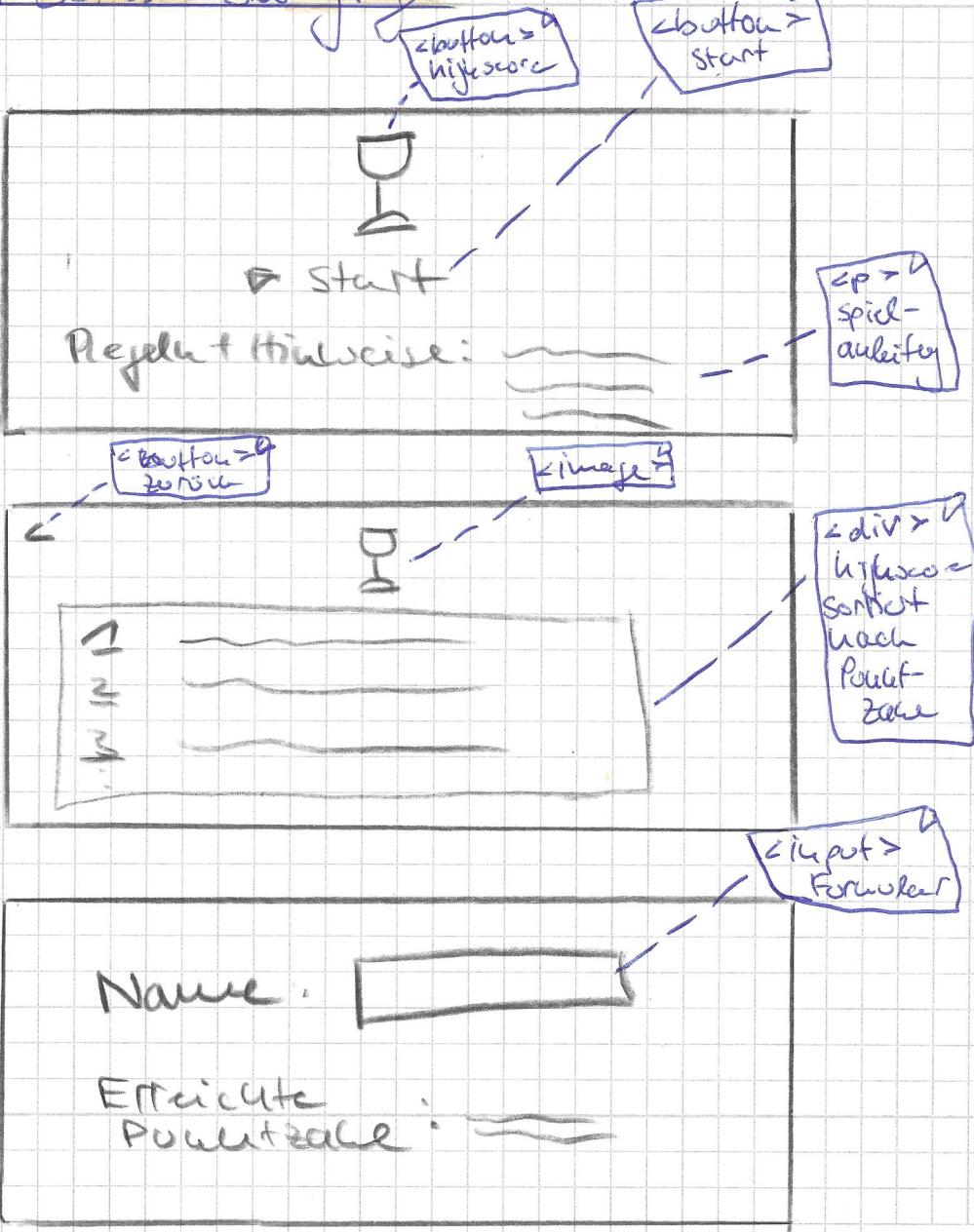
⑤ Snowball

⑥



-2-

UI-Scribble Startig Page



IG

UI - Scribble StartPage II

- Änderungen -

`<px>`

Regeln:

1
2
3

`<button>`

Start

Hightscore

`</button>`

`<div>`
report
(Hightscore)

Erreichte
Punktzahl.

Name:

Prompt

| Informationen, die an den Nutzer gelangen:

- Hightscore-Liste - durch Click auf Button
- Spielanleitung - auf StartPage
- Die eigene Punktzahl - durch einen Prompt am Ende

| Intraktionen der Objekte untereinander:

- Vögel \leftrightarrow Futter

L \rightarrow Wenn Futter gegessen wird \rightarrow 3 Vögel hin.

Die, die vorbei gehen bleiben auch stehen für eine bestimmte Zeit

- Vögel \leftrightarrow Schneeball

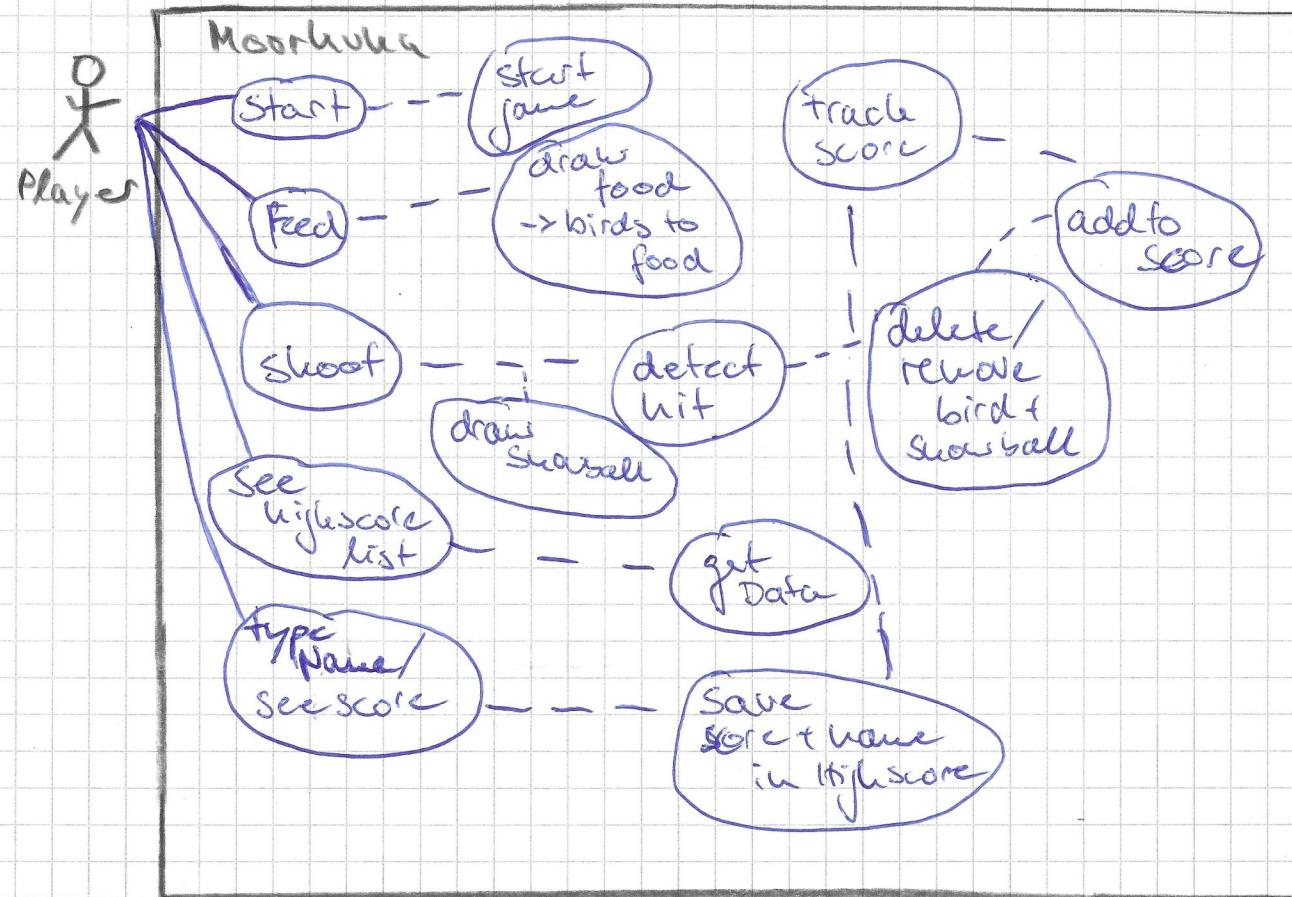
L \rightarrow Wenn Vogel getroffen \rightarrow Vogel + Schneeball weg.

L \rightarrow Erhöhung der Punktzahl um 10

L \rightarrow Schneeball trifft erst nach Vögeln

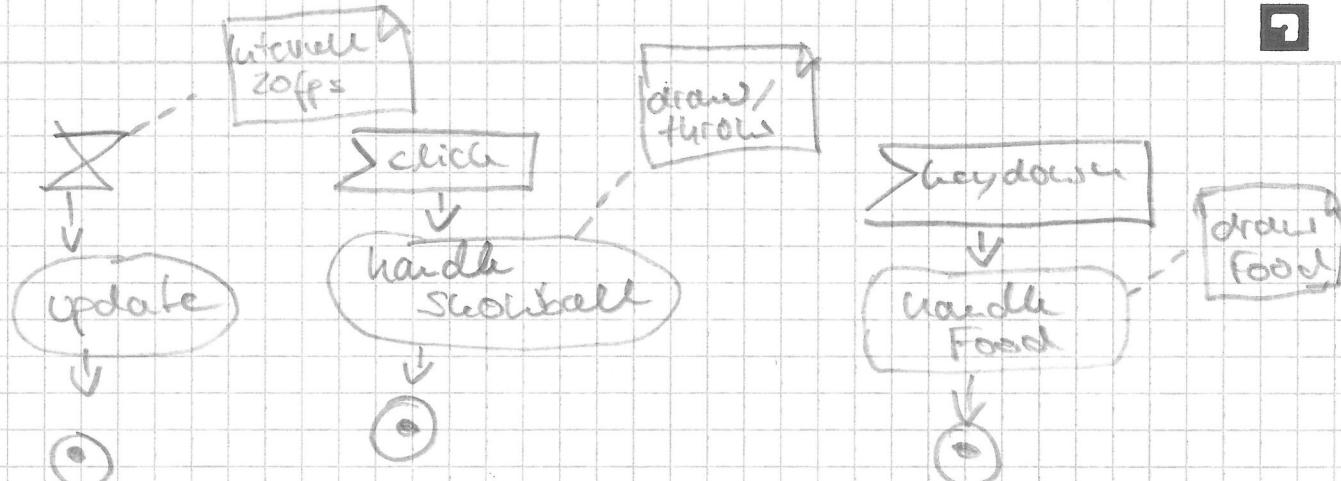
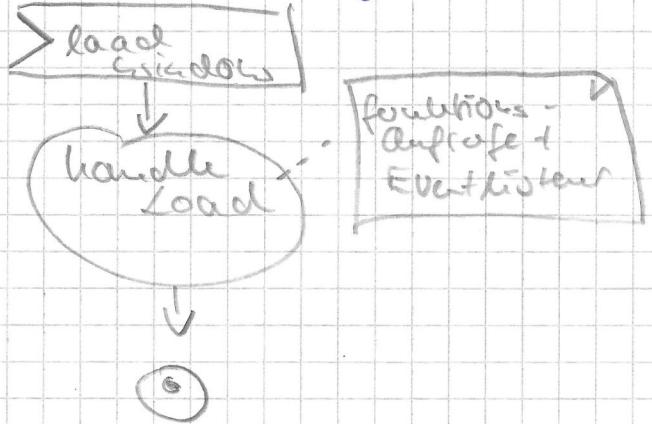


Auwendungsfalldiagramm I

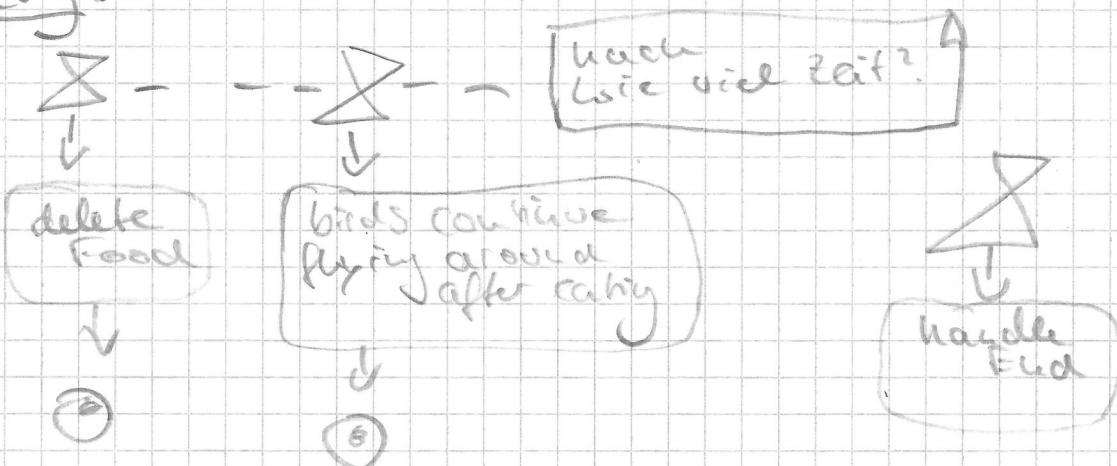


2 Technische Analyse

Activity Diagram I



Ergänzung:



Noch offene Testfragen:

- Vcl20 per V Schneeball
- Vögel zermahlen Futter
- Vögel auffreffen
- Daf funktioniert
- Ende

IG

Vergely Schneeball



Ticker, der immer
verrigert wird
wird neue Frame
-> in update



mit abfrage beim
Vogel auswerfen



a-> Ticker: weiter

-> Vogel Array durchgehen

-> checken, ob Vogel der gerade im
Array drin ist getroffen wurde



Weiter ja
diesen löschen
& score
anzufügeln



Weiter
weil
weiter
im Array

in Schneeball
klasse

Wie werden Vogel
getroffen?



checken, ob Schneeball-
position = Vogel Position



if Noe + Ticker des Schnee-
balls auf bestimmter Zahl



Vogel + Schneeball
aus Array löschen



Vögel zum Futter



Leertrichtung
geändert



Ziel - Start
-> Futter Position -
Vogel Position



3 Vögel zum
Futter



Das Ende



nach ablauf
gewisser Zeit



Öffne Prompt



Nähe eintragen ->
Aktuelle Score

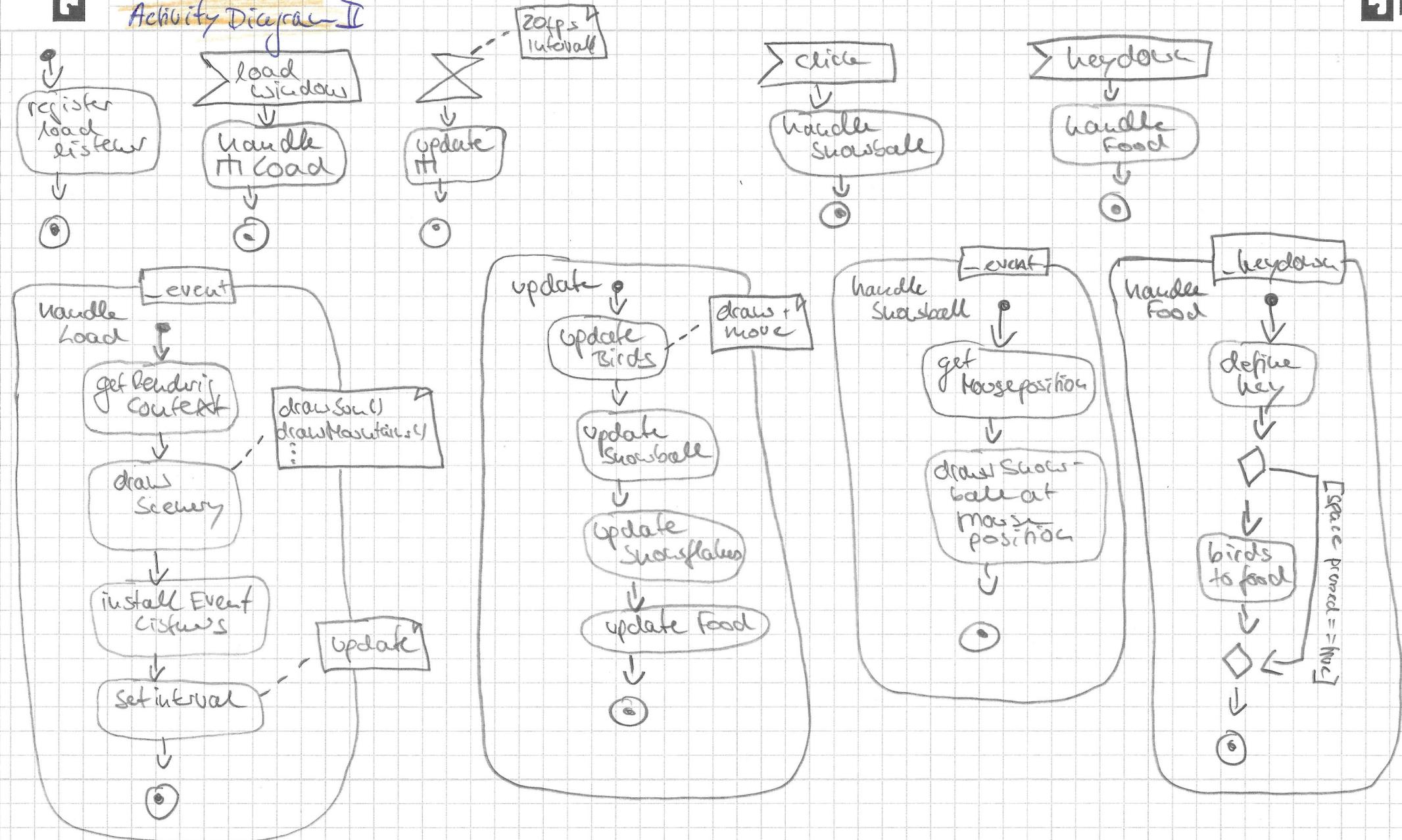


Als Service
Schricken
zur Speicherung
in DB



-7-

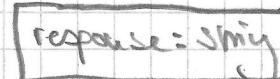
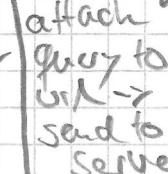
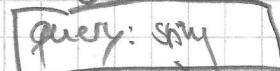
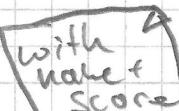
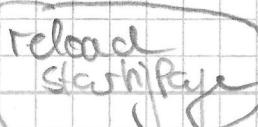
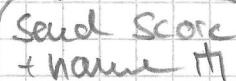
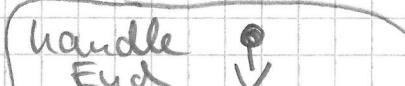
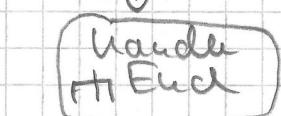
Activity Diagram II



IG

X

[20 sek]



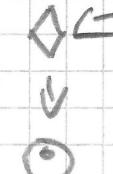
Vögl Wölfer
Übung Schneeball:

Snowball . update

time -= 0.5

collision

[time = 45]



Collision

if player = 0

if bird & snowball

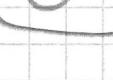
if player collision

if checkHit

if true

delete bird + Snowball

add to score



checkHit: boolean

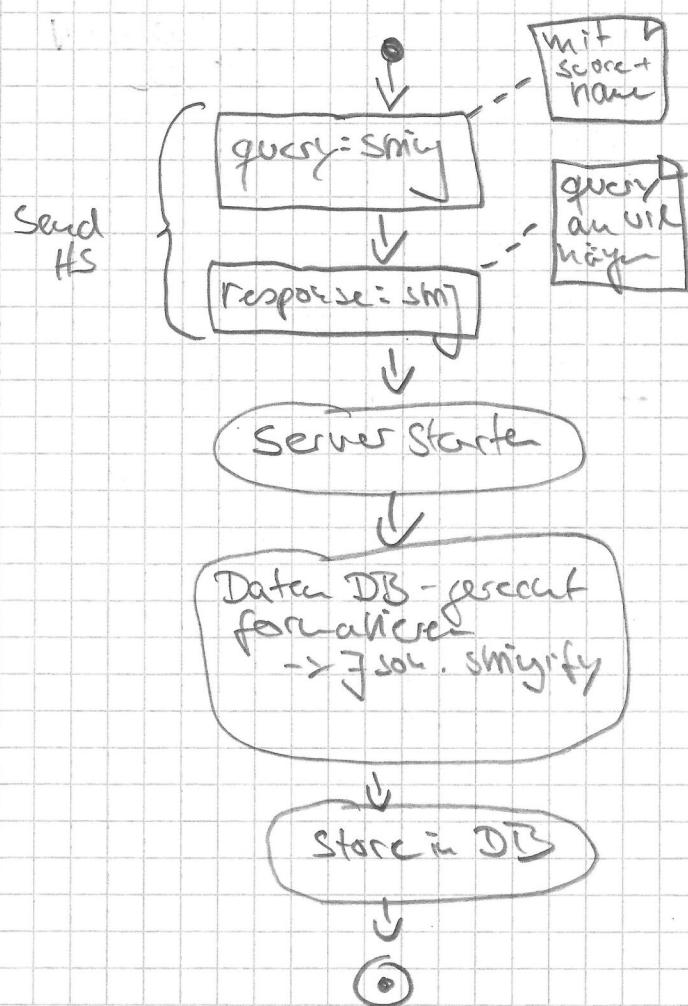
return true



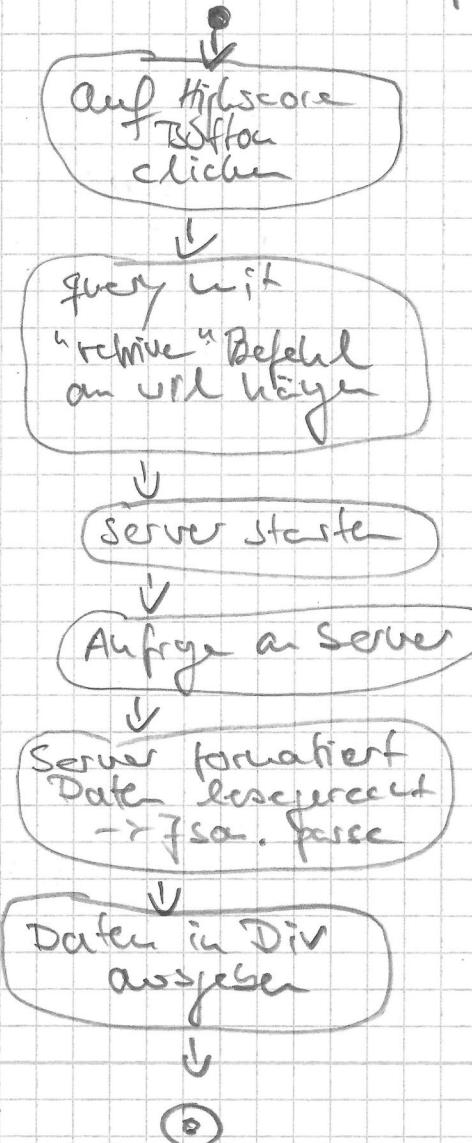
return false

Activity Diagram - DB -

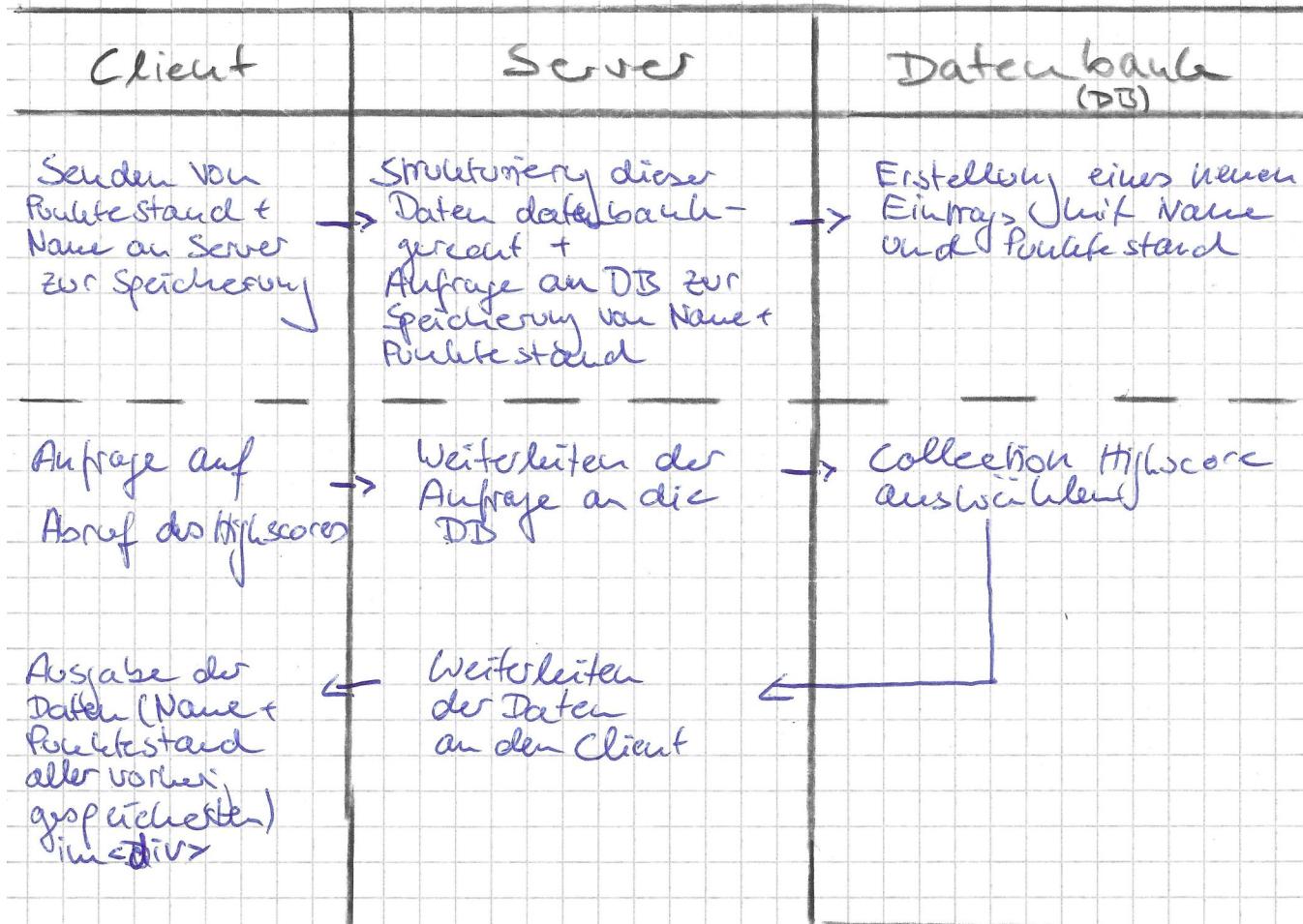
Score + Name an DB schicken



Daten von DB anfragen



Datenübertragung des Aktivitätsdiagramms



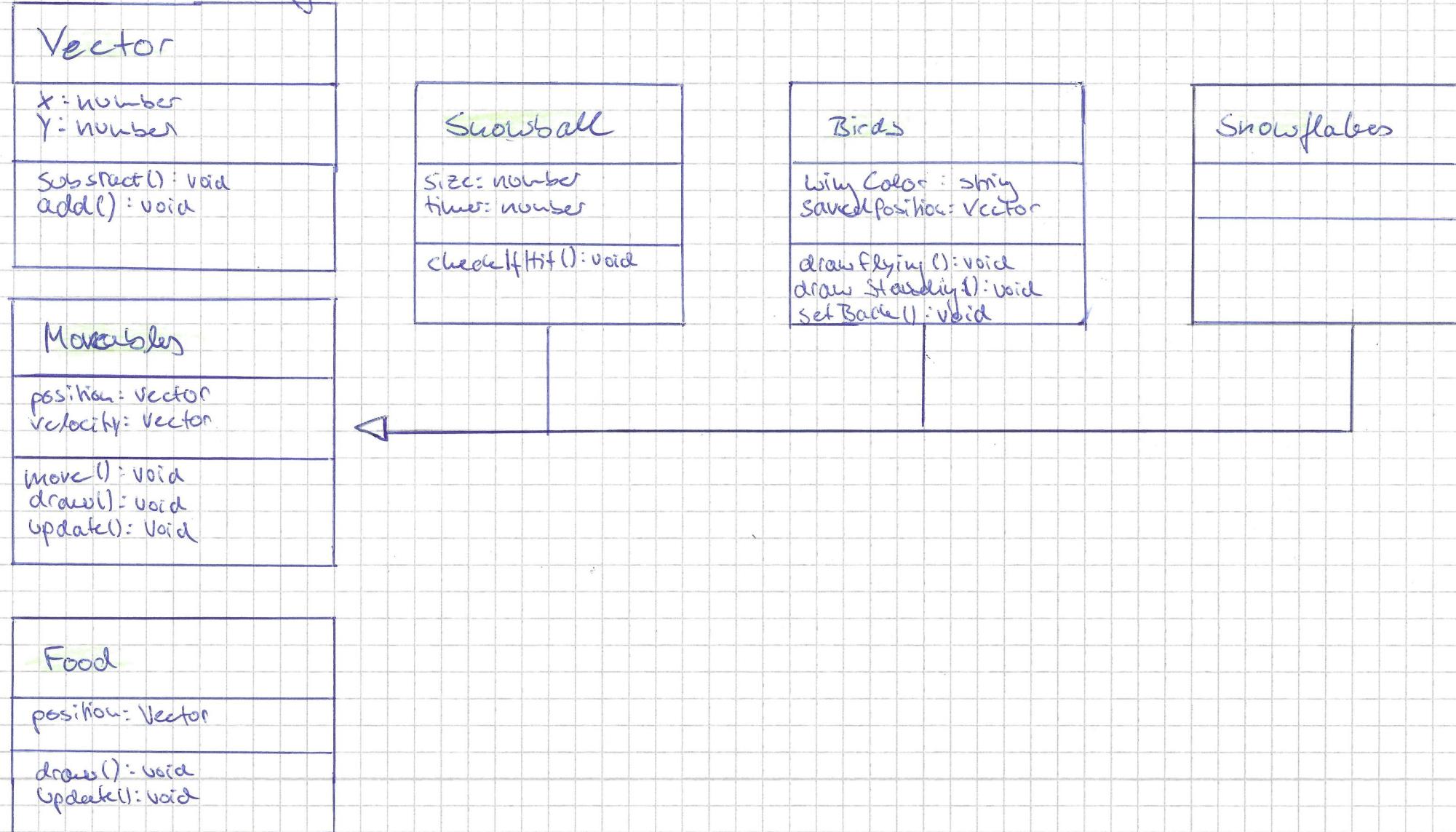
Anwendungsfall:

Spieler ist am Ende des Spiels
und gibt im Prompt seine
Name ein → und bestätigt

Anwendungsfall:

Nutzer klickt auf den Highscore-
Button

Klassendiagramm Final



Auritory Installation

I MongoDB: New Cluster
Name: mooskuh
Collection: Highscores

↳ Add New user → Link in Server.ts
database url!

II Heroku: open app
URL → link

mit GitHub verknüpfen
→ automatic deploy